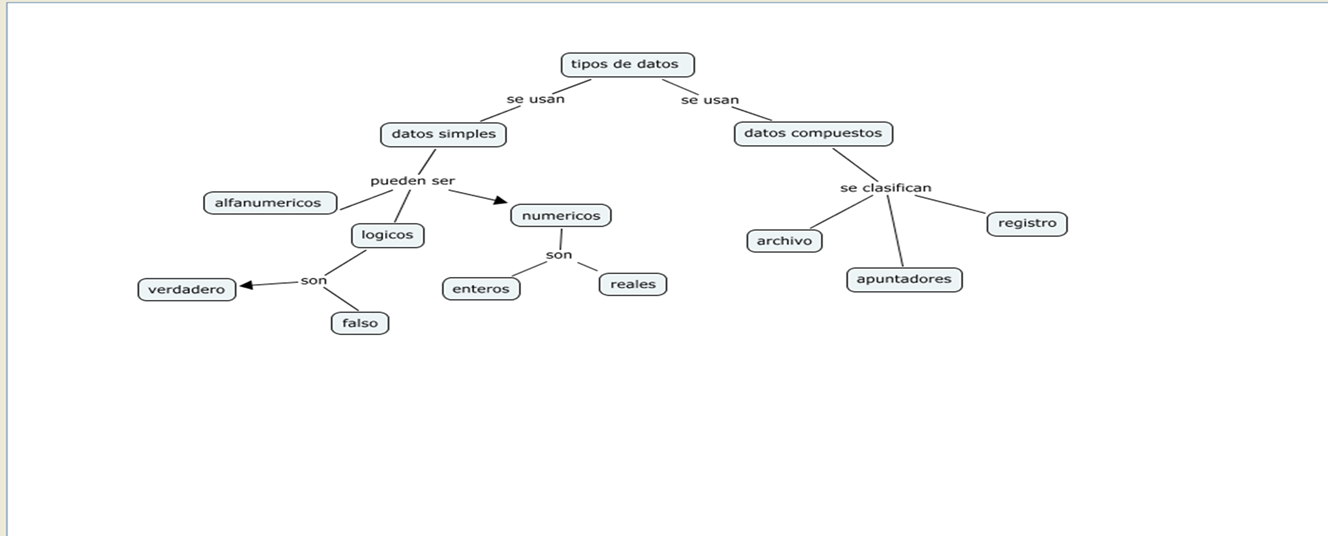
25/04/12

Daniela Sánchez Páez

Maricela lozano Perea

Conceptos básicos sobre algorítmica.

****

**Que es un identificador**

Es un nombre introducido por el programador para hacer referencia a una variable función o tipo de dato definido por el usuario

**Que representa un identificador**

Son símbolos léxicos que nombran entidades del lenguaje

**Cuáles son las reglas para formar un identificador**

Debe comenzar con una letra (A a Z, mayúsculas o minúsculas) y no deben contener espacios en blanco.

Letras, dígitos y caracteres como la subraya ( \_ ) están permitidos después del primer carácter.

La longitud de identificadores puede ser de varios caracteres. Pero es recomendable una longitud promedio de 8 caracteres.

El nombre del identificador debe dar una idea del valor que contiene.

**Ejemplos de un identificador**

Algunos identificadores válidos que pueden ser definidos por el programador son:

Numero

dia\_del\_mes

PINGUINO1

\_ciudad

Z

Ejemplo 2: Los siguientes identificadores no son válidos por incumplir la segunda regla:

123

\_DÍA

Numero\*

Lugar de nacimiento

Año

Ejemplo 3: Los siguientes identificadores no pueden ser definidos por el programador:

int

Char

Int y char son identificadores predefinidos (ya existen en lenguaje C), por tanto, no pueden ser definidos por el programador, en cumplimiento de la tercera regla.

**Que una constante**

Es un valor que no puede ser alterado durante la ejecución de un programa es una constante y la

Constante corresponde a una longitud fija de un área reservada en la memoria principal del ordenador, donde el programa almacena valores fijos

**EJ:**

Char c = 'X'; // X es una constante tipo char

Const int X = 10; // X es un tipo int-constante

EJ:

Cuando se quería utilizar una constante había que utilizar un define (4.9.10b) en la forma:

# define PI = 3.13159;

# define G = 9.80665;

**EJ:**

Directas si están directamente expresadas. Por ejemplo:

Const float pi = 3.14159;

**EJ:**

Expresiones (3.2.3a) si su valor está implícito en una expresión. Por ejemplo:

Const float k = pi \* 2;

**EJ:**

En const int x = 5; no está garantizado que el compilador le asigne a x un Lvalue, es decir, un espacio en memoria determinado (que se permita modificar o no su valor es otra cuestión).

**Que es una variable**

Es una característica (magnitud, vector o número) que puede ser medida, adoptando diferentes valores en cada uno de los casos de un estudio**.**

**EJ:** es un símbolo que permite identificar a un elemento no especifix es una variable del universo {1, 3, 5, 7}. Por lo tanto, x puede ser igual a cualquiera de los recién mencionados valores, con lo cual es posible reemplazar a X por cualquier número impar que sea inferior cado dentro de un determinado grupo

**EJ:** http://www.ejemplode.com/images/uploads/image/variables.jpg

**EJ:**

Cualquier número real para x nos da otro real para y, **luego el dominio es R, pero como x2 > O, el recorrido** serán números positivos o cero.

y= 3 - 2x/ (x-1)(x-2)

**EJ:**

**Sea la regla de correspondencia r: r(x) = x2 + 2x**

**r(2) = 22 + 2(2)=8 (2, 8)**

**r(a) = a2 + 2a, (a, a2 + 2a)**

**r(a+ 1)=(a+ 1)2 + 2(a+ 1)**

**= a2 + 2a + 1 + 2a + 2**

**=a2+4a+3, (a+l,a2+4a+3)**

**EJ:** El dominio, el recorrido y la regla de correspondencia definen una función; antes dijimos la función definida por 2x+y=3 ¿nos estarnos contradiciendo? no es así realmente, lo que sucede, es que por razones prácticas el dominio y el recorrido no se explican y sólo se da la regla de correspondencia, considerando que de antemano se aclaró que trabajamos en el campo de os y unieron reales, de manera que quien "lee" la regla de correspondencia puede de ahí, determinar el dominio y el recorrido, aunque esto no siempre es fácil. En estos casos e dice que ambos, dominio y recorrido están implícitos en la regla de correspondencia.

2x+y=3 ó y=3-2x